

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-267960
 (43)Date of publication of application : 28.08.2000

(51)Int.Cl. G06F 13/00
 H04L 12/56

(21)Application number : 11-076377

(71)Applicant : LION CORP
 CSK CORP

(22)Date of filing : 19.03.1999

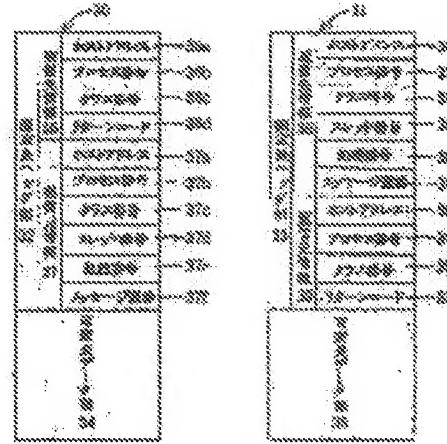
(72)Inventor : SATO TOYOKI
 KOBAYASHI AKITO

(54) METHOD FOR COMMUNICATING PACKET BETWEEN PLURAL PROCESSES AND PACKET COMMUNICATION EQUIPMENT

(57)Abstract:

PROBLEM TO BE SOLVED: To easily transmit/receive a packet between plural processes and to easily prepare a program in an inter-process packet communication equipment using a prescribed packet for the transmission/reception of a message between processes.

SOLUTION: A prescribed transmission packet 30 is transmitted by a class instance of a transmission source process and a transmitted process receiving the packet 30 accesses an instance specified by the packet 30, and when a replay is requested by the packet 30, sends a prescribed reply packet 31. A replied process receiving the packet 31 accesses the class instance of the transmission source process specified by the packet 31.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's

(00) 日本国特許庁 (JP)

02 会開特許公報 (A)

(11)特許出願公開番号

特開2000-287960

(P2000-287960A)

(6)公開日 平成12年9月29日 (2000.9.29)

(5)Int.Cl.*

G 0 6 F 13/00
H 0 4 L 12/00

翻訳記号

3 5 3

F 1

G 0 6 F 13/00
H 0 4 L 13/00

テレコ-ム(翻)

3 5 3 A 6 B 0 8 9
1 0 2 A 8 K 0 3 0
9 A 0 0 1

審査請求 失却求 請求項の数 5 CL (全 10 頁)

(21)出願登録番号

特願平11-76377

(22)出願日

平成11年3月19日 (1999.3.19)

(71)出願人 0000000763

ライオン株式会社

東京都新宿区西新宿1丁目3番7号

(71)出願人 000131201

株式会社シーエスケイ

東京都新宿区西新宿2丁目6番1号

(72)発明者 伊藤 義之

東京都新宿区本郷一丁目3番7号 ライオ
ン株式会社内

(74)代理人 130078835

弁護士 村田 審雄

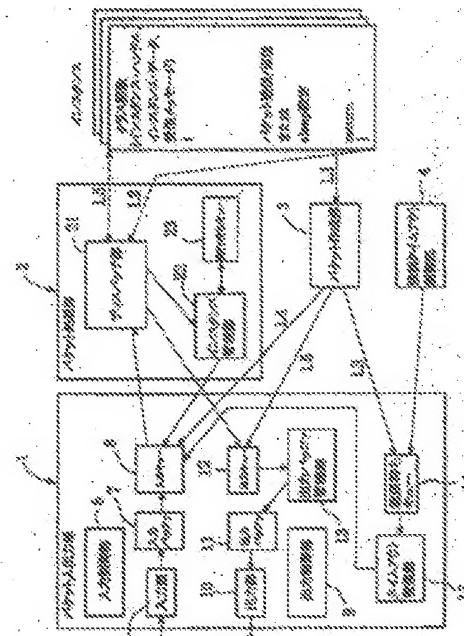
最終頁に続く

(34)【発明の名稱】 プロセス間のパケット送信方法及びパケット送信装置

(57)【要約】

【課題】 プロセス間のメッセージの送受信に用意のパケ
ットを使用するプロセス間のパケット送信装置において、
プロセス間で容易にパケットの送受信を実現し、かつ
プログラムの作成を容易にすることを目的とする。

【解決手段】 発信元プロセスのクラスインスタンスに
おいて、所定の送信パケット30を発信し、該送信パケ
ット30を受信した発信先プロセスにおいて、該パケ
ット30により特定されるインスタンスを呼び出し。該
送信パケット30により送信が要求されている場合に、
所定の送信パケット31を発信し、該送信パケット31を
受信した発信先プロセスにおいて、該パケット31に
より特定される発信元プロセスのクラスインスタンスを
呼び出すことを特徴とする。



【特許請求の範囲】

【請求項1】 プロセス間のメッセージの送受信に所定のパケットを使用するプロセス間のパケット通信方法であって、

発信元プロセスのクラスインスタンスにおいて、発信先のプロセス及びクラスを特定する識別データと、該発信元プロセスのクラスインスタンスを特定する情報と、返信を要求するか否かを示す情報を、を含む発信パケットを発信し、

前記発信パケットを受信した受信先プロセスにおいて、前記発信パケットにより特定される発信先クラスのインスタンスを呼び出し、前記発信パケットにより返信が要求されている場合には、前記発信元プロセスのクラスインスタンスを特定する情報を含む返信パケットを返信し、前記返信パケットを受信した受信先プロセスにおいて、前記返信パケットにより特定される発信元プロセスのクラスインスタンスを呼び出す、

ことを特徴とするプロセス間のパケット通信方法。

【請求項2】 プロセス間のメッセージの送受信に所定

のパケットを使用するプロセス間のパケット通信方法であって、

発信元プロセスのクラスインスタンスにおいて、発信先のプロセス及びクラスを特定する識別データと、該発信元プロセスのクラスインスタンスを特定する情報と、返信を要求するか否かを示す情報を、を含む発信パケットを発信し、該発信パケットを返信待ちパケットとして保持すると共に発信先プロセスからの返信を得たずに他の処理を行い、

前記発信パケットを受信した受信先プロセスにおいて、前記発信パケットにより特定される発信先クラスのインスタンスを呼び出し、前記発信パケットにより返信が要求されている場合には、前記発信元プロセスのクラスインスタンスを特定する情報を含む返信パケットを返信し、前記返信パケットを受信した受信先プロセスにおいて、該返信パケットの識別データと一致する識別データを有する前記返信待ちパケットにより特定される発信元プロセスのクラスインスタンスを呼び出す、

ことを特徴とするプロセス間のパケット通信方法。

【請求項3】 プロセス間のメッセージの送受信に所定のパケットを使用するプロセス間のパケット通信方法であって、

発信元プロセスのクラスインスタンスにおいて、発信先のプロセス及びクラスを特定する識別データと、該発信元プロセスのクラスインスタンスを特定する情報を、返信を要求するか否かを示す情報を、返信を受信した後に前記発信元プロセスにおいて行う処理を特定する情報をと、を含む発信パケットを発信し、該発信パケットを返信待ちパケットとして保持すると共に発信先プロセスからの返信を得たずに他の処理を行い、

前記発信パケットを受信した受信先プロセスにおいて、

前記発信パケットにより特定される発信先クラスのインスタンスを呼び出し、前記発信パケットにより返信が要求されている場合には、前記発信元プロセスのクラスインスタンスを特定する情報を、返信を受信した後に前記発信元プロセスにおいて行う処理を特定する情報をと、を含む返信パケットを返信し、

前記返信パケットを受信した返信先プロセスにおいて、該返信パケットの識別データと一致する識別データを有する前記返信待ちパケットにより特定される発信元プロセスのクラスインスタンスの処理を行う、

ことを特徴とするプロセス間のパケット通信方法。

【請求項4】 複数の接続相手とのメッセージの送受信

に所定のパケットを使用するパケット通信装置であつて、

パケット通信の接続相手ごとに、該接続相手から入力されるパケットを格納する入力バッファと、該接続相手に出力するパケットを格納する出力バッファと、を有するチャネルを備えてなると共に、

接続相手から接続要求があつた場合に、前記チャネルのうちの空きチャネルを該接続相手に割り当てる接続管理手段と、

入力可能状態である前記チャネルの入力バッファにパケットを入力する入力制御手段と、

出力可能状態である前記チャネルの出力バッファからパケットを出力する出力制御手段と、

を備えてなることを特徴とするパケット通信装置。

【請求項5】 前記チャネルにおいて入力されたパケ

ットを一時的に退避させる入力キューと、

前記チャネルから出力するパケットを一時的に退避さ

せる出力キューと、

前記出力キュー内のパケットに対応するチャネルが空

状態の場合に、該出力キューに入っているパケットを該チャネルの出力バッファに入力する出力バッファ管理手段と、

を備えてなることを特徴とする請求項4に記載のパケ

ット通信装置。

【発明の詳細な説明】

【0001】

【発明の属する技術】本発明は、プロセス間のメッセージの送受信に所定のパケットを使用するプロセス間のパケット通信装置において、プロセス間で容易にパケットの送受信を実現し、かいプログラムの作成を容易にすることができるプロセス間のパケット通信方法及びパケット通信装置に関するものである。

【0002】

【従来の技術】同一コンピュータ内又は異なるコンピュータ内に分散配置されたオブジェクト間の通信においては、R.P.C (Remote procedure Call) をベースとした技術を用いてシステム構築が行われている。各種ラン

ダクション、モニタ、オブジェクト・リクエスト、プロ

3

一方等にもRPC又はIRPCと同様の技術が使われ、メッセージによる処理の呼び出しを行っている。また、上記メッセージを複数のコンピュータに送信あるいは複数のコンピュータから取得する場合には、スイッチングハブ等に代表される集配接続装置や、各コンピュータからのメッセージを順に入力し、それぞれメッセージの送信先に配信するシステムが用いられる。

【 0003 】

【 説明が解決しようとする問題】しかしながら、上述のRPCを用いたシステム開発においては、共通言語で記述されたインターフェース部分のプログラムをアプリケーション側の言語に適合させるため、プリコンパイルが必要であり、必ずしも使いやすい開発環境とはいえないかった。また、送信したメッセージに対する返信を持つ場合の処理の方針として両端型と非両端型があるが、両端型では、返信メッセージを受信するまで処理が止まってしまうこと、あるいは処理を行なうことができない。一方、非両端型では、返信メッセージを受信するまで処理が止まってしまうことはないが、返信されたメッセージを処理するためのプログラムが必要になり、また返信されたメッセージを処理すると共に他の処理も行うためには、複雑なプログラミングが要求される。ここで、非両端型において返信されたメッセージを処理すると共に他の処理を行なうプログラミングを簡易に実現する方法として、マルチプロセス又はマルチスレッドによる方法がある。しかし、マルチプロセスによる方法ではコンピュータ資源を余計に使ってしまうという問題があり、マルチスレッドによる方法ではメモリの持続管理や通信のための接続管理等が必要になり、プログラムが複雑になってしまいうる問題がある。

【 0004 】また、上述のようにメッセージを複数のコンピュータに送信する場合にはスイッチングハブ等を用いると、2つのコンピュータ間で通信が行われている間は、その他のコンピュータに対する通信が行われず、その他のコンピュータにおける処理は待ち状態になる。更に、各コンピュータからのメッセージを順に入力し、それぞれメッセージの送信先に配信するシステムでは、1のコンピュータからのメッセージを全て入力してから次のコンピュータからのメッセージを入力し、1のコンピュータへのメッセージを全て出力してから次のコンピュータへのメッセージを出力するという方法のため、大きなメッセージを入出力する場合、あるいは通信速度が遅いコンピュータと通信する場合には、他のコンピュータにおける通信が待たされ、システム全体の性能が低下することになる。

【 0005 】本発明は、このような従来の問題点を認めてなされたもので、プロセス間のメッセージの送受信に所定のパケットを使用するプロセス間のパケット通信装置において、プロセス間で容易にパケットの送受信を実現し、かつプログラムの作成を容易にすることができる

プロセス間のパケット通信方法及びパケット通信装置を提供することを目的とする。

【 0006 】

【 課題を解決する手段】このような従来のプロセス間のパケット通信方法における問題点を解決するために、請求項1に記載の本発明は、プロセス間のメッセージの送受信に所定のパケットを使用するプロセス間のパケット通信方法であって、発信元プロセスのクラスインスタンスにおいて、発信先のプロセス及びクラスを特定する識別データと、該発信元プロセスのクラスインスタンスを特定する情報と、返信を要求するか否かを示す情報と、を含む発信パケットを発信し、前記発信パケットを受信した受信先プロセスにおいて、前記発信パケットにより特定される発信先クラスのインスタンスを呼び出し、無記載パケットにより返信が要求されている場合に、前記発信元プロセスのクラスインスタンスを特定する情報を含む返信パケットを返信し、前記返信パケットを受信した発信先プロセスにおいて、前記返信パケットにより特定される発信元プロセスのクラスインスタンスを呼び出すことを特徴としている。

【 0007 】また、請求項2に記載の本発明は、発信元プロセスのクラスインスタンスにおいて、発信先のプロセス及びクラスを特定する識別データと、該発信元プロセスのクラスインスタンスを特定する情報と、返信を要求するか否かを示す情報と、を含む発信パケットを発信し、該発信パケットを返信待ちパケットとして保持すると共に発信先プロセスからの返信を持たずに他の処理を行い、前記発信パケットを受信した発信先プロセスにおいて、前記発信パケットにより特定される発信先クラスのインスタンスを呼び出し、前記発信パケットにより返信が要求されている場合に、前記発信元プロセスのクラスインスタンスを特定する情報を含む返信パケットを返信し、前記返信パケットを受信した受信先プロセスにおいて、該返信パケットの識別データと一致する識別データを有する前記返信待ちパケットにより特定される発信元プロセスのクラスインスタンスを呼び出すことを特徴とする。

【 0008 】さらに請求項3に記載の本発明は、発信元プロセスのクラスインスタンスにおいて、発信先のプロセス及びクラスを特定する識別データと、該発信元プロセスのクラスインスタンスを特定する情報を、返信を要求するか否かを示す情報を、返信を受信した後に前記発信元プロセスにおいて行なう処理を特定する情報を、を含む発信パケットを発信し、該発信パケットを返信待ちパケットとして保持すると共に発信先プロセスからの返信を持たずに他の処理を行い、前記発信パケットを受信した発信先プロセスにおいて、前記発信パケットにより特定される発信先クラスのインスタンスを呼び出し、前記発信パケットにより返信が要求されている場合に、前記発信元プロセスのクラスインスタンスを特定する情報を

5

と、遅延を受信した後に前記遅滞先プロセスにおいて行う処理を特定する情報と、を含む遅延パケットを遅延し、前記遅延パケットを受信した遅滞先プロセスにおいて、該遅延パケットの識別データと一致する識別データを有する前記遅滞待ちパケットにより特定される発信元プロセスのクラスインスタンスの処理を行うことを特徴とする。

【 0 0 0 9 】 また、請求項4に記載の本発明は、複数の接続相手とのメッセージの送受信に所定のパケットを使用するパケット通信装置であって、パケット通信の接続相手ごとに、該接続相手から入力されるパケットを格納する入力バッファと、該接続相手に出力するパケットを格納する出力バッファと、を有するチャンネルを備えてなると共に、該接続相手から接続要求があった場合に、前記チャンネルのうちの空きチャンネルを該接続相手に割り当てる接続管理手段と、入力可能状態である前記チャンネルの入力バッファにパケットを入力する入力制御手段と、出力可能状態である前記チャンネルの出力バッファからパケットを出力する出力制御手段と、を備えてなることを特徴として構成されている。

【 0 0 1 0 】 さらに請求項4に記載の本発明は、前記請求項4に記載の本発明において、前記チャンネルにおける入力されたパケットを一時的に退避させる入力キューと、前記チャンネルから出力するパケットを一時的に退避させる出力キューと、前記出力キュー内のパケットに対応するチャンネルが空状態の場合に、該出力キューに入っているパケットを該チャンネルの出力バッファに入力する出力バッファ管理手段と、を備えてなることを特徴として構成されている。

【 0 0 1 1 】

【 発明の実施の形態】 以下、本発明の第一の実施形態について図面を参照して詳細に説明する。図1は、本実施形態におけるプロセス間パケット通信装置の一例の構成を示す図、図2は、本実施形態におけるパケットのデータ構造を示す図、図3はパケット通信の手順を示す図である。本実施形態におけるプロセス間パケット通信装置は、同一又は異なるコンピュータのプロセス内のオブジェクト間で所定のパケットを用いてメッセージを交換するものである。ここで各オブジェクトは、プロセス内のクラスとして定義される。プロセス内には1つ又は複数のクラスが定義され、1つのクラスが1つのサービスを提供する。また各クラスは各種の処理を実行する実数等で定義される。各クラスには1つのクラス番号が付与され、コンピュータのがストアドレッス、プロセス番号、及びクラス番号を特定することによりクラスが一意に識別される。また各クラスにおいてはインスタンスが生成され、このインスタンスに対してクラス毎にスレッド番号が付与される。従って、クラス番号とスレッド番号をからなるインスタンス・ハンドルによってインスタンスが一意に識別される。更に各クラスには、インスタンスが

存在する間にインスタンス固有のデータを保存するためのメモリ領域(インスタンス・データ領域、略称せず)を実現することができる。このインスタンス・データ領域については後述する。

【 0 0 1 2 】 図1に示すように本実施形態におけるプロセス間パケット通信装置は、パケット入出力部1、パケット処理部2、パケット作成部3、接続タイムアウト制御部4とを各プロセスごとに備えてなる。パケット入出力部1は、パケットの入出力処理とパケット送信のタイムアウトを制御するものであり、入力制御部5、入力部6、入力バッファ7、入力キュー8、出力制御部9、出力部10、出力バッファ11、出力キュー12、出力バッファ管理部13、遅滞待ちキュー14、及びタイムアウト監視部15を備えてなる。パケット入出力部1の入力制御部5は、入力可能状態となっている入力部6を呼び出し、該入力部6にデータを入力する。またパケット入出力部1の入力部6は、一定時間内に入力したデータ又は一定量のデータを入力バッファ7に入力する。そして、1つのパケットの入力が完了すると、入力バッファ7内のデータを入力キュー8に出力し、入力部6は次のパケットの入力可能状態となる。パケット入出力部1の出力制御部9は、出力バッファ11にデータがあるときに、出力可能状態となっている出力部10を呼び出しして出力バッファ11内のデータを出力する。またパケット入出力部1の出力バッファ管理部13は、出力バッファ11にデータがないときに、出力キュー12に置かれたパケットを取り出して出力バッファ11に入れる。パケット入出力部1の出力部10は、一定時間内に出力するデータ又は一定量のデータを出力バッファ11から取り出し、出力する。そして、1つのパケットの出力が完了すると、出力バッファ13は空状態となり、出力部10は出力待ち状態となる。またパケット入出力部1のタイムアウト監視部15は、遅滞待ちキュー14に移動された遅滞待機中のタイムアウト時間を監視する。

【 0 0 1 3 】 またパケット処理部2は、パケット入出力部1において送受信されるパケットに基づいてクラスインスタンスの生成と消滅を制御するもので、ディスクバッヂ部21、インスタンス管理部22、及び実行待ちキュー23を備えてなる。パケット処理部2のディスクバッヂ部21は、パケット入出力部1の入力キュー8からパケットを取り出し、該パケットによって特定される遅滞先のプロセス内のクラスのインスタンスを起動又は消滅させる。また、インスタンス管理部22は、後述する最大スレッド数に基づいてインスタンスの多量度を管理する。パケット作成部3は、プロセス間の通常に用いられるパケットの遅滞先ヘッダ部と可変長部を作成する。パケットの遅滞先が遅滞元と異なる他のプロセスであるときは、作成したパケットを出力キュー12に入れる。一方、パケットの遅滞先が、遅滞元と同じプロセスであるときは、作成したパケットを直接入力キュー8に入れ、

7

送信する無数を省いている。また遅延タイムアウト制御部4は、遅延待ち情報を強制的にタイムアウトさせるものである。

【0014】次に、本実施形態にて使用するパケットについて説明する。図2(ア)は本実施形態にて使用する発信パケットのデータ構成を示す図、(イ)は遅延パケットのデータ構成を示す図である。本実施形態にて使用するパケットとしては、メッセージを発信する発信パケット30と該メッセージに対する遅延メッセージを遅延する遅延パケット31の2種類がある。これら発信パケット、遅延パケットともに、図2に示すように発信先(あるいは遅延先)と発信元(あるいは遅延元)を特定する固定長ヘッダ部3.2、3.3とメッセージを搭載した可変長データ部3.4、3.5とから構成される。図2(ア)に示すように、遅延パケット3.0の固定長ヘッダ部3.2は、遅延先情報3.6、発信元情報3.7から構成される。この発信先情報3.6及び発信元情報3.7にはそれぞれコンピュータのホストアドレス3.6a、プロセス番号3.6b、クラス番号3.6cからなる識別データが含まれ、これらの識別データにより発信先あるいは発信元のプロセス及びクラスが特定される。また発信先情報3.6は、上記識別データに加えてリターンコード3.6dを含んでおり、このリターンコード3.6dには、パケット遅信中にエラーが起きたときにエラーコードが設定される。発信パケット3.0における発信元情報3.7は、上記識別データ3.7a～3.7cに加えて発信元のクラスインスタンスのスレッド番号3.7d、媒體番号3.7e、及びメッセージ番号3.7fを含んでいる。このスレッド番号3.7dと上記発信元の識別データ3.7a～3.7cとともに上り、発信元のクラスインスタンスを一意に識別することができる。また図2(イ)に示すように、遅延パケット3.1における遅延先情報3.8は、遅延データ3.8a～3.8c、スレッド番号3.8d、処理番号3.8e、及びメッセージ番号3.8fを含んでおり、遅延元情報3.9には、識別データ3.9a～3.9c及びリターンコード3.9dを含んでいる。即ち、遅延パケット3.1の固定長ヘッダ部3.3は、遅延パケット3.0の発信先情報3.6と発信元情報3.7を、遅延元情報3.9と遅延先情報3.8に入れ替えたものである。

【0015】次に、本実施形態によるプロセス間パケットと遅延接続におけるパケット遅信の手順について説明する。まず、遅延パケットの発信の手順について説明する。あるクラスから他のクラス(発信元クラスと同一のプロセス内のクラスであってもよいし、異なるプロセス内のクラスであっててもよい。また、他のクラスとあるが、発信元クラスと同一のクラスであっててもよい。)にメッセージを遅信する際には、パケット作成部3を呼び出し(し1)、発信する遅延パケットを作成する。この遅延パケットの固定長ヘッダ部3.2の発信先情報3.6には、発信先クラスのホストアドレス、プロセス番号、及

8

リクラス番号が設定され、発信元情報3.7には、発信元クラス自身のホストアドレス、プロセス番号、クラス番号、及び該クラスインスタンスのスレッド番号等が設定される。この遅延パケットの固定長ヘッダ部3.2により、発信先のクラス及び発信元のクラスインスタンスを一意に識別することができる。

【0016】ここで遅信先のクラスインスタンスが発信先クラスからの遅信を要求しない場合には、発信パケットの固定長ヘッダ部3.2の発信元情報3.7の処理番号3.7eに0を設定する。一方、発信先クラスからの遅信を要求する場合には、上記処理番号3.7eに0以外の値を設定する。即ち、この発信元情報3.7の処理番号3.7eは遅信を要求するか否かを示す指標として機能する。

【0017】発信パケットの発信元情報3.7の処理番号3.7eに0以外の値を設定し、発信先クラスからの遅信を要求する場合には、遅延パケットを発信したクラス番号から以外の遅信にて遅る(し2)。このとき遅信元のクラスインスタンスはそのまま待機し、遅延待ち状態となる。この遅延待ち状態となる際に、発信した遅延パケットのコピー(遅延待ちパケット)及び遅延待ち状態のタイムアウト時間が、遅信待ち情報としてパケット入出力部1の遅延待ちキュー1.4に保存される(し3)。なお、この遅延待ち情報のタイムアウト時間は、各パケットごとに異なる値を設定することができる。

【0018】ここで、パケット入出力部1のタイムアウト監視部1.5は、この遅延待ち情報のタイムアウト時間を監視する。タイムアウトした場合は、タイムアウト・エラー・コードをリターンコード3.9dに設定した遅延パケットが、パケット作成部3により作成される。その後、このタイムアウト・エラーの遅延パケットを入力キュー8に入れる(し4)と共に、該遅延待ち情報を削除される。なお、このタイムアウト・エラーの遅延パケットを(出力キュー1.2ではなく)直接入力キュー8に入れることで、遅延パケットを遅信する無数を省いている。また、上記パケット入出力部1の遅延待ちキュー1.4に保存された遅延待ち情報は、該タイムアウト制御部4によってクラスインスタンス単位あるいはクラス単位又は全てを強制的にタイムアウトさせることができる。このときには、タイムアウト・エラー・コードの替わりに任意のリターンコード3.9dを設定することができる。なお、遅延パケットの発信先を非ULIクラスに設定すると、実際には遅延パケットは発信されず、上記遅延待ち情報のみがパケット入出力部1の遅延待ちキュー1.4に保存される。そして、該遅延待ち情報をタイムアウト時間後に対応するクラスインスタンスが呼び出されることとなり、クラスインスタンスを一定時間休止(stay)させ、一定時間間隔での繰り返し処理が可能となる。

【0019】一方、発信パケットの発信元情報3.7の処理番号3.7eに0を設定し、発信先クラスからの遅信を

9

要求しない場合には、遅滞パケットを発信したクラス番号から遅滞IDにて戻り(L2)。パケット処理番号のディスパッチ番号2 1 がこの発信元のクラスインスタンスを継承させる。パケット作成旗3 により作成された遅滞パケットは、パケット入出力部1 の出力キュー1 2 に入れられ、出力バッファ1 1 、出力部1 0 を介して発信側のプロセスに送信される。上述のように、返信を要求する場合は、クラスインスタンスの中からパケットを発信する必要があるが、返信を要求しないのであれば、発信元のクラスインスタンスを特定するクラス番号とスレッド番号は特別の処理を持たないため、クラス番号とスレッド番号に0以外の値を設定すれば、クラスインスタンスの外からパケットを発信することも可能である。

【 0 0 2 0 】次に、上記発信パケットの受信及び遅滞パケットの返信の手順について説明する。発信パケットが発信先プロセスで受信されると、発信パケットの固定長ヘッダ部3 2 の発信先情報3 6 で特定した発信先クラスのインスタンスがパケット処理番号2 のディスパッチ番2 1 により生成される。そして該クラスインスタンスが呼び出され、発信パケットのメッセージが渡される(L5)。なお受信メッセージを受け取ったクラスインスタンスは、1つの遅延スレッドとして起動され、インスタンスを示すハンドルを引数にしてクラス番号が呼ばれる。

【 0 0 2 1 】ここで受信した発信パケットの発信元番号3 7 の処理番号3 7 e が0以外の値の場合、即ち発信元のクラスインスタンスが返信を要求する場合には、パケット作成旗3 を呼び出し(L1)、返信する遅滞パケットを作成する。この遅滞パケットの固定長ヘッダ部3 3 の遅滞元情報3 9 には受信した発信パケットの発信先情報3 6 が設定され、遅滞先情報3 8 には受信した発信パケットの発信元情報3 7 が設定される。これによりメッセージを発信したクラスインスタンスに遅滞パケットが返送されることとなる。遅滞パケットと同様に、作成された遅滞パケットは、パケット入出力部1 の出力キュー1 2 に入れられる(L6)、出力バッファ1 1 、出力部1 0 を介して返信先のプロセスに返信される。

【 0 0 2 2 】一方、受信した遅滞パケットの発信元情報3 7 の処理番号3 7 e が0の場合、即ち発信元のクラスインスタンスが返信を要求しない場合には、上記遅滞パケットは返信されない。なお遅滞パケットを受信した発信先プロセスにおいて、遅滞パケットの固定長ヘッダ部3 2 の発信先情報3 6 で特定したクラスが登録されていない場合は、その旨のエラーコードをリターンコード3 9 d に設定した遅滞パケットがパケット作成旗3 により作成され、該遅滞パケットが発信元クラスに返送される。

【 0 0 2 3 】次に遅滞パケットの受信の手順について説明する。遅滞パケットが発信元プロセスで受信されると、受信された遅滞パケットの遅滞先情報3 8 のクラス

10

10

番号3 8 c 、スレッド番号3 8 d 、メッセージID遅滞3 8 e と上述のパケット入出力部1 の遅滞待ちキュー1 4 に保存された遅滞待ちパケットの遅滞元情報3 7 のクラス番号3 7 c 、スレッド番号3 7 d 、メッセージID遅滞3 7 e を比較し、これらすべてが一致する遅滞待ちパケット(即ち、遅滞パケットに対応する遅滞待ちパケット) を検索する。対応する遅滞待ちパケットが見つかれば、該遅滞待ちパケットの固定長ヘッダ部3 2 の発信元情報3 7 で特定されるクラス、即ち発信パケットを発信したクラスインスタンスが再度呼び出され、該遅滞待ちパケットに対応する遅滞待ち情報を削除される。このとき遅滞パケットの遅滞先情報3 8 の処理番号3 8 e (値はすれば、発信パケットの発信元情報3 7 の処理番号3 7 e) に対応する処理が行われる。従って発信パケットの発信元情報3 7 の処理番号3 7 e は、上述のように返信を要求するか否かを示す情報として識別すると共に、遅滞したときに実行処理を特定する情報としても機能する。

10

【 0 0 2 4 】更に上記処理番号は、クラスインスタンス内で、遅滞パケットにて呼び出されたのか、遅滞パケットにて呼び出されたのかを識別するためにも使用することができます。つまり、クラスインスタンスに関する処理番号を遅滞用処理番号と処理識別用処理番号の2つに分け、遅滞用処理番号には、発信パケットの発信元情報3 7 の処理番号3 7 e を入れ、処理識別用処理番号には、発信パケットのときに3 0 を、遅滞パケットのときには発信先情報3 8 の処理番号3 8 e を入れる。遅滞先情報3 8 の処理番号3 8 e は0になることがないため、処理識別用処理番号が0のときは、発信パケットで呼び出されたことになり、0以外のときは遅滞用パケットで呼び出されたことになる。なお、このクラスインスタンスの呼び出しに際し、上述のインスタンス・データ領域に保存したデータを使用して前回呼び出し時の処理を継続することができる。

10

【 0 0 2 5 】この場合において、あるクラスインスタンスが遅滞待ち状態であっても、他のクラスインスタンスに対する遅滞パケットが受信されると対応するそのクラスのインスタンスが起動されるが、このクラスインスタンスが終了するか又は遅滞待ち状態になれば、該遅滞待ち状態のクラスインスタンスに対する遅滞パケットの受信により該遅滞待ちのクラスインスタンスが再度起動されることになる。一方、対応する遅滞待ちパケットが見つかなければ、警告メッセージをログに出力し受信した遅滞パケットは放棄される。なお、受信したパケットが発信パケットあるいは遅滞パケットのハンドルであるからは、パケットの固定長ヘッダ部のスレッド番号の設定によって判別される。即ち、パケットを受信したプロセスは、受信したパケットの発信先情報3 6 (あるいは遅滞先情報) にスレッド番号が設定されていないならば発信パケットを受信したと判断し、発信先情報3 6 (あるいは遅滞先情報) にスレッド番号が設定されているならば

10

送信パケットを受信したと判断する。

【 0026】ここでクラスごとの最大スレッド数又は全クラス合計の最大フレッド数を指定することができる。パケット処理部2のインスタンス管理部2.2は、パケットを受信してもクラスインスタンスを作成せず、該受信したパケットをインスタンス管理部2.2内の実行待ちキュー2.3に入れる。この実行待ちキュー2.3に入れられたパケット（実行待ちパケット）は、クラスインスタンスが消滅したときに取り出され、該パケットに対応するインスタンスが生成される。即ち、クラスインスタンスが消滅したときには、そのクラスのスレッド数と全クラスのスレッドの合計が1つ減るので、実行待ちパケットの中で、対応するクラスが最大スレッド数に達しておらず、かつ最初に入った実行待ちパケットが取り出される。このように最大スレッド数を指定可能とすることによつて、クラスごとの多量度又は全クラスの多量度を制御でき、コンピュータ資源の消費及び性能を容易にコントロールすることができる。

【 0027】次に、上述したパケット通信の手順を具体例を挙げて説明する。図3は本実施形態に係るプロセス間パケット通信装置におけるパケット通信の手順を示したものである。以下、図3を参照して説明する。まずプロセス101からプロセス102に発信パケットを発信する（S1）。このときの発信パケットの発信元情報3.7には「プロセス番号=101、発信番号=0」が設定され、発信先情報3.6には「プロセス番号=102、クラス番号=1」が設定されている。このプロセス101のプロセスから発信されたパケットはプロセス管理部100を介してプロセス102に送られる（S2）。なおプロセス管理部100は、プロセスの起動・終了、接続登録・登録・解除の管理や、各プロセスの監視、各プロセスへの通知、各プロセス間におけるパケットの中継等の機能を有するものである。

【 0028】プロセス102では、パケット入出力部1によって上記発信パケットが受信され、パケット処理部2によって、発信パケットの発信先情報3.6によって特定されるクラス番号1のクラスインスタンスが生成される（S3）。このとき発信先情報部番号は0となるので、クラス1内の発信番号0の処理が実行される。なお、通信用処理番号は0であるので発信元のプロセス101には返信しない。上記プロセス102のクラス1内の処理では、更に発信パケットが発信される（S4）。この発信パケットの発信先情報3.6には「プロセス番号=103、クラス番号=3」が設定され、発信元情報3.7には「プロセス番号=102、クラス番号=1、発信番号=10」が設定される。ここで、この発信パケットは「処理番号=10」であり、返信を要するものである。従って返信にて該クラス開設から戻り（S5）、返信待ち状態になる。

【 0029】一方、上記プロセス102から発信された

発信パケットはプロセス管理部100を介してプロセス103のパケット入出力部1により受信される。そしてプロセス103のパケット処理部2によりクラス番号3のクラスインスタンスが生成される（S6）。このクラス内の処理では発信パケットが返信される（S7）。この返信パケットの返信先情報には、受信した発信パケットの発信元情報3.7。即ち「プロセス番号=102、クラス番号=1、発信番号=10」が設定され、返信元情報には、受信した発信パケットの発信先情報3.6、即ち「プロセス番号=103、クラス番号=3」が設定される。このプロセス103から返信された返信パケットは、プロセス管理部を介してプロセス102のパケット入出力部1により受信される。そして、プロセス102のパケット処理部2によりクラス番号1のクラスインスタンスが再度呼び出され（S8）、該クラス1内の処理番号10の処理が実行される。

【 0030】プロセス103のクラス1内の発信番号1の処理においては、プロセス104へ発信パケットを発信する（S9）。このときの発信パケットの発信先情報3.6は「発信番号=0」とされ、返信を持たないので該クラス開設から返信にて戻る（S10）。このように各プロセス間におけるメッセージの送受信が可能となる。

【 0031】次に本発明の第二の実施形態について説明を参照して詳細に説明する。なお特に説明しない部分については上記第一の実施形態と同様である。図4は本実施形態におけるプロセス間パケット通信装置の構成の一例を示す圖である。図4に示すように、本実施形態におけるプロセス間パケット通信装置のパケット入出力には、接続するコンピュータごとに1つの接続チャネルが割りかれている。それぞれのチャネルには、入力端8、入力バッファ7、及び出力バッファ11が設けられる。なお、この入力バッファ7及び出力バッファ11は、1つのパケットの全てを接続できる大きさを持つ。

【 0032】ここで、本実施形態におけるプロセス間パケット通信装置には、本装置への接続要求（又は切断要求）を監視する接続管理部5.0が設けられている。この接続管理部5.0は、接続要求があれば接続し、接続先を各チャネルに割り当てる。逆に、切断要求があれば接続を切断し、チャネルの動作を解消する。そして、接続される相手方のコンピュータからは、一定時間内に入力できたデータ又は一定量のデータ（パケット）がそれぞれ入力され、入力バッファ7に入れられる。また、接続される相手方のコンピュータには、一定時間内に出力できるデータ又は一定量のデータ（パケット）が出力バッファ11から取り出され出力される。1つのパケットの入力が完了すると、入力バッファ7内を入力待ちキュー8に並びし、入力端8は次のパケットの入力待ち状態（入力可能状態）となる。また、1つのパケットの出力が完了すると、出力バッファ11は空状態になる。

13

【 0033 】 パケット処理部2は、入力キューからパケットを取り出し、パケットの固定長ヘッダ部にある発信先情報(送信先情報)に対応する出力チャネル情報を付加して出力キュー12に入れる。出力バッファ管理部13は、出力バッファ11に入っているパケットに対応するチャネルの出力バッファ11が空状態のときは、そのパケットを出力キュー12から取り出して該空状態の出力バッファ11に入れる。出力制御部9は、出力バッファ11(パケットが残っている(出力可能状態)チャネルの出力部10を順次呼び出してデータを出力する。一方、入力制御部6は、入力可能状態となっているチャネルの入力部6を順次呼び出してパケットを入力する。なお、各チャネルには優先順位を付けることができ、特定のチャネルの通信を優先させることができるのである。入力部6と出力部10とを異なる優先順位とすることもでき、また同一の優先順位とすることもできる。

【 0034 】

【 発明の効果】 上述したように本発明は、プロセス間のメッセージの送受信に所定のパケットを使用するプロセス間のパケット通信方法であって、発信元プロセスのクラスインスタンスにおいて、所定の発信パケットを発信し、該発信パケットを受信した発信先プロセスにおいて、発信パケットにより特定されるクラスインスタンスを呼び出し、通信要求されている場合に、所定の返信パケットを返信し、該返信パケットを受信した発信先プロセスにおいて、該返信パケットにより特定されるクラスインスタンスを呼び出すことにより、RPCを用いなくともプロセス間にてメッセージの送受信が可能となるので、プログラムを作成する際にプリコンパイルをする必要がなくなり、容易にプログラムを作成することができるという効果がある。

【 0035 】 また発信元プロセスのクラスインスタンスにおいて、所定の発信パケットを発信し、該発信パケットを返信待ちパケットとして保持すると共に発信先プロセスからの返信を得たずに他の処理を行い、発信パケットを受信した発信先プロセスにおいて、発信パケットにより特定される発信先クラスのインスタンスを呼び出し、発信パケットにより返信が要求されている場合に、所定の返信パケットを返信し、該返信パケットを受信した返信先プロセスにおいて、該返信パケットのデータデータを返信待ちパケットとに蓄へて発信元プロセスのクラスインスタンスを呼び出すことにより、発信パケットに対する返信を得る場合であっても、返信パケットを受信したときに発信が呼び出されるので、返信を得ている間ににおいて他の処理を行うことができ、かつ、このための特別なプログラムを必要としないという効果がある。また、返信パケットには、返信後の処理を特定する情報を含めることとしたので、返信後の処理を容易に特定してプログラムを作成することができるという効果がある。さらにマルチプロセスではなく、1つのプロセス

14

内の機能としてパケットの送受信を行うことができるので、コンピュータの資源を余計に使うことがないという効果がある。また、OSレベルで制御を行うマルチスレッドではなく、開放レベルでマルチスレッドを実現している(開放マルチスレッド)ので、メモリの持続的管理や通信のための接続管理等のプログラムを必要としないという効果がある。

【 0036 】 さらに本発明は、複数の接続相手とのメッセージの送受信に所定のパケットを使用するパケット通信装置であって、パケット通信の接続相手ごとに入力バッファと出力バッファとを有するチャネルと、接続相手から接続要求があった場合に、新規チャネルのうちの空きチャネルを接続相手に割り当てる接続管理手段と、入力可能状態である前記チャネルの入力バッファにパケットを入力する入力制御手段と、出力可能状態である前記チャネルの出力バッファからパケットを出力する出力制御手段と、を備えてなることにより、1つのチャネルに対して1つパケットの入出力が完了する前に、他のチャネルの処理を行うことができ、パケットの大きさ又は通信速度の違いに影響されずに各チャネルに対する平均的な処理速度を保証できるという効果がある。

【 0037 】 またチャネルにおいて入力されたパケットを一時的に蓄積する入力キューと、チャネルから出力するパケットを一時的に蓄積する出力キューと、出力キュー内のパケットに対応するチャネルが空状態の場合に、該出力キューに入っているパケットを該チャネルの出力バッファに入力する出力バッファ管理手段と、を備えてなることにより、各チャネルの入力と出力及び通信処理が相互の処理状態の影響を受けずに動作するので、全体としてのスループットが向上するという効果がある。

【 図面の簡単な説明】

【 図1 】 本発明の第一の実施形態におけるプロセス間パケット通信装置の一例の構成を示す図である。

【 図2 】 本発明の第一の実施形態におけるパケットのデータ構成を示す図で、(a)は発信パケットのデータ構成を示す図、(b)は返信パケットのデータ構成を示す図である。

【 図3 】 本発明の第一の実施形態におけるパケット通信の手順を示す図である。

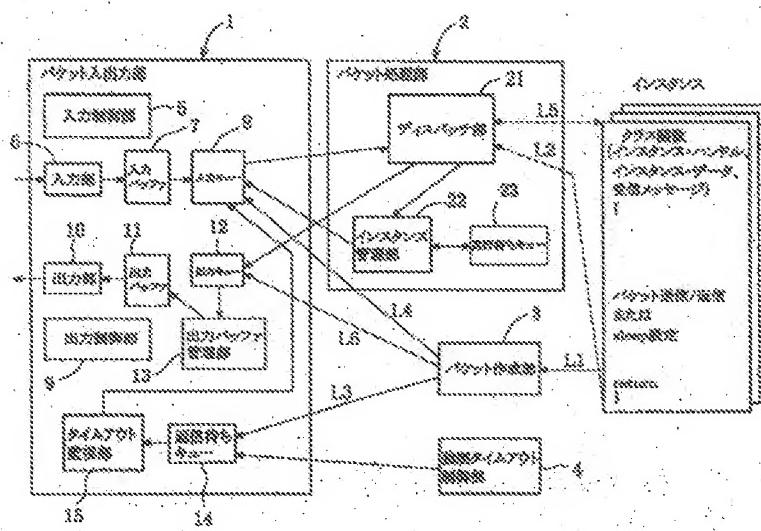
【 図4 】 本発明の第二の実施形態におけるプロセス間パケット通信装置の構成の一例を示す図である。

【 符号の説明】

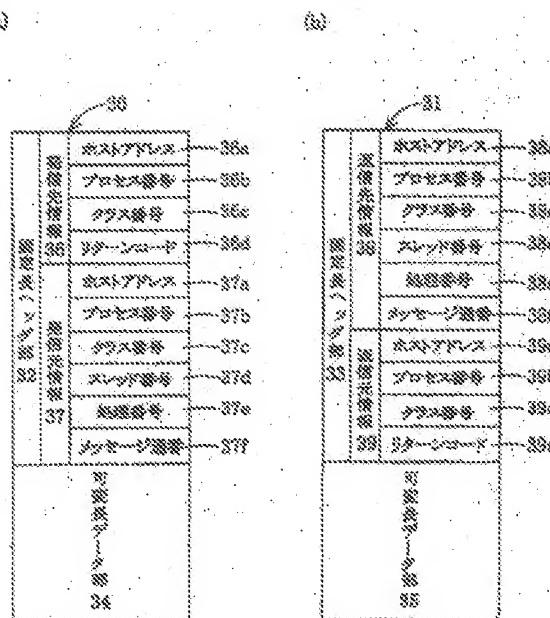
- | | |
|----|----------|
| 1 | パケット入出力部 |
| 5 | 入力制御部 |
| 7 | 入力バッファ |
| 8 | 入力キュー |
| 9 | 出力制御部 |
| 11 | 出力バッファ |

- | | | | |
|-----|-------------|-----|--------|
| 1 2 | 出力キュー | 3 0 | 発信パケット |
| 1 3 | 出力バッファ管理部 | 3 6 | 発信先情報 |
| 2 | パケット送達部 | 3 7 | 発信元情報 |
| 2 1 | ディスクシナプス部 | 3 1 | 迷惑パケット |
| 3 | パケット作成部 | 3 8 | 迷惑先情報 |
| 4 | 強制タイムアウト制御部 | 3 9 | 迷惑元情報 |

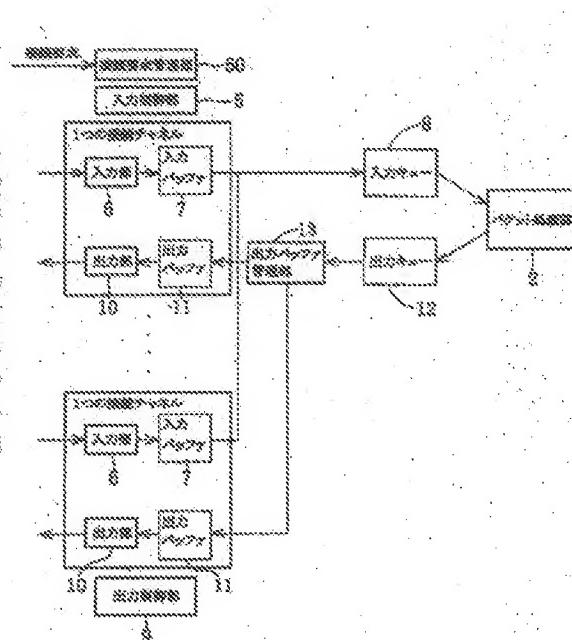
[図1]



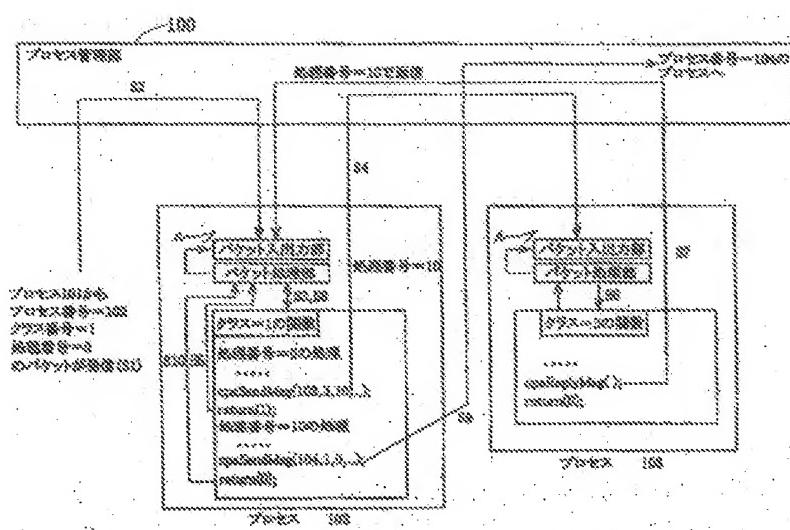
[図2]



[図4]



(22)



フロントページの概念

(72)発明者 小林 明人
東京都新宿区西新宿2丁目6番1号
会社シーエスケイ内

Pターム(参考) 58089 5902 5905 5904 5905 5910
5911 5906 5909 5901
5906 5903 5908 5915 5928
59001 5904 5902 5916 5954